## REMARKS

Claims 38-72 were pending at the time of examination. Claims 38, 41-42, 44, 46-49, 51-54, 58, 60-62, 64-65, 67-69 and 71 have been amended. No new matter has been added. The applicants respectfully request reconsideration based on the foregoing amendments and these remarks.

### Claim Rejections – 35 U.S.C. § 101

Claims 38-72 were rejected under 35 U.S.C. § 101 because they are directed to non-statutory subject matter. With respect to the rejection of claims 38-59, the applicant has amended claim 38 to recite a "computer implemented method" in the preamble. With respect to the rejection of claims 60-72, the applicant has amended the preamble of claim 60 to recite

> "A computer program product, stored on a machine-readable medium, containing computer code for managing resource usage of code downloaded to a computer system, the computer program product including:"

The applicant respectfully submits that independent claims 38 and 60, as amended, and their respective dependent claims, are directed to statutory subject matter and request that the rejection of claims 38-72 under 35 U.S.C. § 101 be removed.

### Claim Rejections – 35 U.S.C. § 102

Claims 38 and 60 were rejected under 35 U.S.C § 102(e) as being anticipated by Czajkowski G. et al. "Internet Servers, Safe-Language Extensions, and Structured Resource Control," Technology of Object-Oriented Languages and Systems, 1999, Proceedings of Nancy, France 7-10 June 1999, Los Alamitos, CA, USA, IEEE Comput. Soc., Us, 7 June 1999, pages 295-304 (hereinafter "Czajkowski").

Czajkowski describes as system for alleviating problems related to resource management in server environments, in particular to ascertain that a request that is received by a server does not consume or monopolize an entire resource in the server. For this purpose Czajkowski uses a "resource account." The resource account "explicitly encapsulates the <u>rights to various computational resources</u>" (Czajkowski, page 5, 4th paragraph). Resource accounts can be shared by multiple threads, if needed. Whenever an attempt is made to charge more resources than allowed by a given resource account, an associated resource overuse callback is raised (Czajkowski, page 5, 5th paragraph). For at least these reasons, it is submitted that the anticipation rejection of claim 38 is unsupported by the cited art and should be withdrawn.

Claim 60 is a *Beauregard* claim corresponding to claim 38. For reasons substantially similar to those set forth above with respect to claim 38, the applicant respectfully contends that the rejection of claim 60 is unsupported by the cited art and should be withdrawn.

## Claim Rejections – 35 U.S.C. § 103

Claims 38-46, 50-58, 60-63, and 66-71 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 5,838,968 to Culbert (hereinafter "Culbert") in view of U.S. Patent No. 6,430,570 to Judge et al. (hereinafter "Judge"). Claims 47-49 and 64-65 were rejected under 35 U.S.C § 103(a) as being unpatentable over Culbert in view of Judge as applied to claims 38, 45, 60 and 63, further in view of U.S. Patent No. 6,182,022 to Mayle et al. (hereinafter "Mayle"). Claims 59 and 72 were rejected under 35 U.S.C § 103(a) as being unpatentable over Culbert in view of Judge as applied to claims 38 and 60, further in view of Applicant's admitted prior art. The applicant respectfully traverses these rejections.

### Culbert does not teach or render obvious related code, associated resource indicators, or tracking of related code only

Claim 38 requires:

> "for each code downloaded to the computer system, **associating a resource indicator with all threads that are executed directly** by the downloaded code **and all threads that are initiated by the downloaded code**, wherein all of the threads that are executed directly by the downloaded code and all threads that are initiated by the downloaded code are defined as a **set of related code**;"

That is, the claim limitation teaches a grouping of threads that are executed directly by downloaded code, and all threads that are initiated by downloaded code. Each such group of threads is referred to as a set of related code. Each set of related code has an associated resource indicator.

The Examiner alleges that this is shown in FIGs. 3 and 4 of Culbert, and further refers to sections of Culbert that discuss resource usage for tasks. The Examiner appears to equate the related code of the applicant's invention with the tasks in Culbert. The

Examiner also appears to equate the "task utilization record" of Culbert with the applicant's resource indicators. This is clear, since the Examiner specifically points to col.3, lines 20-58 of Collins and quotes (and underlines to add particular emphasis) a section stating "…keeping track of actual system resource utilization through periodic measuring by updating <u>the current task utilization record to reflect the consumption of the plurality of system resources</u>…"

The applicant respectfully submits that none of the sections cited by the Examiner discloses the notion of **related code**. In fact, Culbert does not even mention that its tasks may be related.

The applicant also respectfully submits that Culbert does not teach any **resource indicators** associated with the related code, as required by claim 38. Culbert's "task resource utilization records" are specified <u>by the tasks.</u> Each task resource utilization record contains information about the required quantities of resources managed by a resource manager that are required for the particular task to function properly (col. 7 lines 48-51). Furthermore, a task may specify multiple task resource utilization records (col. 7, line 67 – col 8, line 1), which is contrary to the applicants invention, where a particular piece of code is only associated with one resource indicator.

Claim 38 further specifies:

> "updating the resource indicator every time that the set of related code changes its actual collective resource usage of a particular resource so that the resource indicator **only** tracks actual resource usage of the set of **related code**;"

That is, in the applicant's invention, as defined in claim 38, <u>only</u> the set of related code is tracked by the associated resource indicator. Culbert cannot tracking related code only, since Culbert contains no mechanism for identifying related code and associating it with a resource indicator, as was described above. The ability to only track related code advantageously allows implementation of procedures with respect to each set of threads executed or initiated on behalf of each downloaded code when actual resource usage by such related threads exceeds a particular limit.

<u>Culbert does not teach or render obvious updating a resource indicator as specified in claim 38</u>

Claim 38 specifies that the resource indicator is updated "<u>every time</u> that the set of related code <u>changes</u> its actual collective resource usage of a particular resource."

That is, whenever there is a change of actual collective resource usage in a set of related code, the associated resource indicator is updated. The Examiner appears to think that this is shown in Culbert, and quotes parts of col. 7 and 8 of Culbert saying "updates the usage value of global system resource in resource master list by calling an update routine for maintaining current information based on actual resource usage to ensure the maximum number of concurrent tasks to be supported," and "task resources associated with the current run level are updated with actual resource measurements," respectively. The applicant respectfully disagrees. Updating a "usage value of global system resource in resource master list" is in no way similar to updating an associated resource indicator for a set of related code.

Furthermore, other sections of Culbert describe various triggering mechanisms for the update. For example, in Culbert, an "UpdateResourceMeasurement routine is activated by a timer on a periodic basis..." (col. 8, lines 47-48) or in response to a query from a resource manager (col. 10, lines 52-55). The UpdateResourceMeasurement routine "replaces the utilization record...with actual measured resource utilizations" (col. 8, lines 48-50). Thus, the updates in Culbert are "passive" in nature, since they are triggered by external events and require that a resource utilization measurement occurs in response to the triggering event. Claim 38, however, specifies that a resource indicator is updated "every time that the set of related code changes its actual collective resource usage." That is, in the applicant's invention no "actual measured resource utilizations" are needed, since the respective resource indicators are updated every time the resource usage of the related code changes.

Culbert only updates certain types of tasks

Claim 38 requires "for each code downloaded to the computer system, associating a resource indicator with all threads that are executed directly by the downloaded code and all threads that are initiated by the downloaded code..."

That is, regardless of the type of code that is downloaded to the computer system, a resource indicator is associated with the code. Culbert, on the other hand, defines three classes of tasks: error intolerant, error-tolerant realtime, and non-realtime (see Culbert, col. 8, lines 19-20), and states the "error intolerant tasks never have their resource utilization records updated with actual use" (see Culbert, col. 8, lines 58-59). Thus, since the actual collective resource usage of the error intolerant tasks is never considered in Culbert, there would be no update if the error intolerant tasks were to change their resource usage. In the applicant's invention, as defined by claim 38, the resource indicators will be updated for each code downloaded to the computer system, and not only for a subset of codes or tasks. Thus, even in this aspect the applicant's invention is different from Culbert.

Combining Culbert with Judge does not cure the deficiencies of Culbert

The teachings of Judge do not remedy the deficiencies of Culbert in anticipating or rendering the applicants' invention obvious. The Examiner cited Judge because "Culbert, however, did not specify the tasks (related codes) are downloaded to the system" (Office action, page 6, second paragraph). Judge discloses resource management in a Java system, primarily with respect to memory usage, but fails to teach or suggest tracking resource usage with a resource indicator for all threads that are executed or initiated for each code downloaded to a computer system, in the manner claimed. In Judge, "If, during the execution of an application...the JVM. 22 runs out of memory, an OutOfMemoryError error is generated" (Judge, col. 8, lines 1-3). The applicant respectfully disagrees with the Examiner's assertion that "It is obviously for an ordinary skill in the art to recognize that the memory usage has been tracked for the executed applications," and submit that what is tracked in Judge is a global memory usage, which can be tracked without tracking the usage of individual applications. There is no suggestion in Judge as to how to track the memory usage of individual applications, or as to how to track a set of related code, as required by applicant's claim 38.

In order to establish a *prima facie* case of obviousness, the Examiner must show a motivation to combine Culbert and Judge. Even if Judge shows that code can be downloaded, nothing in either Culbert of Judge suggests a desire to combine the two patents. The fact that both Culbert and Judge relate to alternative ways of managing low-memory conditions in computer devices or systems, both of which are different from the method disclosed in claim 38, is not sufficient as a motivation to combine the two documents. Neither is saying that "the motivation for rejection is found in the knowledge generally available to one of ordinary skill in the art," as the Examiner stated in section 45 of the most recent Office Action. Furthermore, the Examiner needs to show a reasonable expectation of success, which the Examiner has failed to do since he has not shown how someone skilled in the art would use Judge to cure the above mentioned deficiencies of Culbert. Finally, the combination of the references must teach or suggest all the claim limitations. Even if it were possible to combine Culbert and Judge, the combination still would not teach, for example, the limitation of "updating the resource indicator every time that the set of related code changes its actual collective resource usage of a particular resource so that the resource indicator only tracks actual resource usage of the set of related code," as neither Culbert nor Judge features related code, resource indicators, or does any updates in the manner described in the limitation. For at least these reasons, the rejection of claim 38 is unsupported by the art and should be withdrawn.

Claim 60 is a *Beauregard* claim corresponding to claim 38. For reasons substantially similar to those set forth above with respect to claim 38, the applicants respectfully contend that the rejection of claim 60 is unsupported by the cited art and should be withdrawn.

### The rejections of the dependent claims

The Examiner's rejections of the dependent claims are also respectfully traversed. However, to expedite prosecution, all of these claims will not be argued separately. Claims 39-59 and 61-69, respectively, each depend directly from independent claims 38 and 60 and, therefore, are respectfully submitted to be patentable over cited art for at least the reasons set forth above with respect to claims 38 and 60.

Further, the dependent claims require additional elements that when considered in context of the claimed inventions further patentably distinguish the invention from the cited art. For example, claims 41 and 61 further specify how associating/dissacociating of related code and resource indicators occur; claim 44 specifies that the disassociation is done through a garbage collection procedure, claims 46, 58 and 71, respectively, specify what happens when the resource indicator exceeds a particular level, claims 57 and 70 further require that "determining which threads are to be defined as the set of related code based on which threads are assigned to a same protection domain." Finally, claims 59 and 72 require that "the computer system is integrated with a set top box or a navigational system." The cited references, alone or in combination, fail to teach or suggest any of these limitations.

### Conclusion

The applicant believes that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the

Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Fredrik Mollborn

Fredrik Mollborn
Reg. No. 48,587

P.O. Box 70250
Oakland, CA  94612-0250
(650) 961-8300